CAN'T USE

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06N

A2

(11) International Publication Number: **WO 00/70556**

(43) International Publication Date: 23 November 2000 (23.11.00)

(21) International Application Number: PCT/US00/13823

(22) International Filing Date: 19 May 2000 (19.05.00)

(30) Priority Data:
60/134,793 19 May 1999 (19.05.99) US

(71) Applicant: WHITEHEAD INSTITUTE FOR BIOMEDICAL RESEARCH [US/US]; Nine Cambridge Center, Cambridge, MA 02142 (US).

(72) Inventors: REN, Bing; 39 Springfield Street, Somerville, MA 02143 (US). YOUNG, Richard; 216 Highland Street, Weston, MA 02493 (US). YOUNG, Peter; 48 Lowell Street, Somerville, MA 02143 (US).

(74) Agent: RODRIGUEZ, Michael, A.; Testa, Hurwitz & Thibeault, LLP, High Street Tower, 125 High Street, Boston, MA 02110 (US).

(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

Without international search report and to be republished upon receipt of that report.

(54) Title: A METHOD AND RELATIONAL DATABASE MANAGEMENT SYSTEM FOR STORING, COMPARING, AND DISPLAYING RESULTS PRODUCED BY ANALYSES OF GENE ARRAY DATA

(57) Abstract

A method and system for analyzing data over a network are described. A Web server communicates with a storage system that stores genomic information in a database. Client systems connect to the Web server over a network, such as the Internet, using standard Web protocols (e.g., HTTP). The Web server sends Web pages to the client through which pages the user of the client can load genomic information into the database. The client user obtains the genomic information for uploading from genomic samples of organisms hybridized to chips or arrays. With the database populated with genomic information, the client user interactively selects and performs an analysis on selected samples over the network. The result produced by the analysis is a list of genes or a list of gene lists that becomes part of the database. These gene lists or lists of gene lists can then be compared with other previously stored lists or with user–generated and/or user–selected gene lists. Accordingly, subsequent users of the database can review the research performed by others, and incorporate that research into their own research.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

DID: <WO___0070556A2_I_>

A METHOD AND RELATIONAL DATABASE MANAGEMENT SYSTEM FOR STORING, COMPARING, AND DISPLAYING RESULTS PRODUCED BY ANALYSES OF GENE ARRAY DATA

Related Application

This application claims the benefit of the filing date of copending U.S. Provisional Application, Serial No. 60/134,793, filed May 19, 1999, entitled "Relational Database Management System For Gene Array Data," the entirety of which provisional application is incorporated by reference herein.

Background of the Invention

Array-based expression analysis tools permit the simultaneous measurement of RNA expression levels for all or part of the genome of an organism. Arrays, or "expression chips", that probe every ORF (open reading frame) in the yeast genome, as well as for several other organisms, are now commercially available. Chips probing expression levels of up to 10,000 human genes and ESTs (expressed sequence tags) are also available. The accessibility of parallel expression analysis has ushered in a new era of genetic discovery, where the full genetic behavior of an organism is measurable in parallel. This widely applicable technology is being applied to problems in yeast biology, functional genomics, drug discovery, and other domains.

Despite the great promise that expression profiling holds for biology research, anyone attempting to use array technology quickly discovers that the ability to produce biological data does not imply an ability to interpret that data. Consequently, management and interpretation of the massive data sets produced by expression analysis tools have become a bottleneck in biological research. Techniques used to analyze expression data, which range from pencil and paper to computerized spread sheets, do not provide an adequate means for solving the problems

presented by massive data sets: for example, filtering noise, comparing across data sets, annotating entire genomes, measuring experimental error, and extracting meaningful information from as many as 100,000 data points. Thus, there is a need for data analysis tools that enable researchers to extract information about individual genes across specific conditions as well as integrate large amounts of data to provide an overall picture of expression remodeling under various experimental conditions.

Summary of the Invention

An object of the invention is to automate many of the processes necessary for analyzing data such as for example genomic information obtained from chips or gene arrays. Such processes include loading data sets, rescaling data from different arrays so the data can be compared, data management, and analysis. Another object is to provide data visualization tools that facilitate the interpretation of the results of analyses. Still another object of the invention is to enable researchers to compare different samples in the database, without diminishing the capability of the researchers to learn as much from individual experiments as possible.

To achieve these and other objects, a method and relational database management system for storing, comparing, and displaying results produced by analyses of gene array data are provided. A Web server communicates with a storage system that stores genomic information in a database. Client systems connect to the Web server over a network, such as the Internet, using standard Web protocols (e.g., HTTP). The Web server sends Web pages to the client through which pages the user of the client can load genomic information into the database. The client user obtains the genomic information for uploading from genomic samples of organisms hybridized to chips or arrays. With the database populated with genomic information, the client user interactively selects and performs an analysis on selected samples over the network. The result produced by the analysis is a list of genes or a list of gene lists that becomes part of the

database. These gene lists or lists of gene lists can then be compared with other previously stored lists or with user-generated and/or user-selected gene lists. Accordingly, subsequent users of the database can review the research performed by others, and incorporate that research into their own research.

In one aspect, the invention features a method for analyzing data. The method comprises providing data and rescaling the data to produce rescaled data. The rescaled data may be stored in the same database as the sample result. The rescaled data is associated with a pre-selected set of parameters. A sample set is generated from the associated rescaled data. Analysis is performed on the sample set to produce a sample result, and the sample result is stored in a database. The stored sample result is associated with a prior result. The prior result can be a sample result previously stored in the database, a user-generated result, or a user-selected result.

In one embodiment, the stored sample result is a list of lists. Each list in the list of lists is a list of genes. In another embodiment, the stored sample result is a set of bit vectors. In still another embodiment, the associating comprises comparing the sample result with the prior result. The results of associating the stored sample result with prior result may be stored in the database.

In another aspect, the invention features a system for analyzing data. The system includes a calibrator rescaling the data and a pre-selected set of parameters that is associated with the rescaled data. A sample set is generated from the associated rescaled data. An analyzer performs analysis on the sample set to produce a sample result. A database stores the sample result. An associator associates the stored sample result with a prior result. The prior result can be a sample result previously stored in the database, a user-generated result, or a user-selected result.

Brief Description of the Drawings

The invention is pointed out with particularity in the appended claims. The advantages of the invention described above, as well as further advantages of the invention, may be better understood by reference to the following description taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a block diagram of client-server network providing database services according to the principles of the invention;

Fig. 2 is flow diagram of an embodiment of a process in which the client user accesses the database according to the principles of the invention;

Fig. 3 is a screen shot illustrating an embodiment of a graphical user interface presented to the client user for entering sample data into the database of the invention;

Fig. 4 is a screen shot illustrating an embodiment of a graphical user interface presented to the client user for performing a rule-based analysis on a set of samples;

Fig. 5 is a screen shot illustrating an embodiment of a graphical user interface presented to the client user for performing comparisons between sample results and/or user-selected or generated categories;

Fig. 6 is a block diagram of an embodiment of a schema of the database of Fig. 1; and

Fig. 7 is a screen shot illustrating an embodiment of a graphical user interface presented to the client user for reporting the results of a search for a particular gene in the sample results stored in the database.

Detailed Description

FIG. 1 shows a computing system (client) 10 in communication with a computing system (server) 20 over a network 30. The server 20 is in communication with a storage system 40 providing storage for genomic information and, in accordance with the principles of the invention, storage for the results of analyses performed on the genomic information. It is to be

understood that more clients and servers than those shown can be connected to the network 30. Although shown in Fig. 1 as separate systems, in another embodiment the client 10 and server 20 can be the same machine.

The client 10 can be any personal computer (e.g., 286, 386, 486, Pentium, Pentium II), thin-client device, Macintosh computer, Windows-based terminal, Network Computer, wireless device, information appliance, RISC Power PC, X-device, workstation, mini computer, main frame computer, or other computing device that has a graphical user interface. Windows-oriented platforms supported by the client 10 can include Windows 3.x, Windows 95, Windows 98, Windows NT 3.51, Windows NT 4.0, Windows CE, Windows CE for Windows Based Terminals, Macintosh, Java, and Unix. The client 10 includes conventional hardware for supporting a display screen, a keyboard, memory, a processor, and an input/output device (e.g., a mouse).

The client 10 also has software including browser software 12, e.g., Microsoft Internet Explorer™ produced by Microsoft Corporation of Redmond Washington. The browser software 12 provides a graphical user interface to the server 20. Through the Web browser, the client 10 develops and submits search requests for retrieving data from the storage system 40. In general, the user of the client formulates queries of the storage system 40, using the keyboard and the input device to point and click on graphical buttons, pull down menus, scroll bars, etc., that are then submitted to the server 20 over the network 30

Server 20 includes the hardware necessary for running software to access information in the storage system 40 in response to client user requests, and for providing an interface for transmitting information to the client 10. In one embodiment, the server 20 operates as a Web server 32, supporting the World Wide Web protocol (e.g., HTTP protocol) for providing page

data to the client 10, maintaining Web pages, processing URLs, and controlling access to other portions of the network 30 (e.g., workstations, storage systems, printers) or to other networks. In one embodiment, the server 20 is a 233 MHz Pentium II running on a Windows NT 4.0 workstation. In another embodiment that improves multi-user performance, the server 20 is a Ultra-4 Sparc workstation running the Solaris 2.6 operating system with four 400 MHz processors and 1 GB of RAM (produced by Sun Microsystems).

As shown, the server 20 includes the World Wide Web server 32, a World Wide Web interface 34, and a database management system (DBMS) 36. The Web interface 34 includes the executable code necessary for generating queries that access information in the storage system 40 (e.g., database language statements such as Standard Query Language (SQL) statements). The Web interface 34 also includes Web applications written in PL/SQL, Perl and Java. On Web application enables the client user to directly upload genome expression data files into the storage system 40 (hereafter called the loader 35). Other the Web applications provide a Web interface to the storage system 40 and perform data analysis such as normalization and comparisons between unlimited number of experiments and functional categorization of an organism's genes.

In general, the database management system (DBMS) 36 serves as a Web-based search engine that enables the client user to search for any number of genes according to user-specified key words in names or gene description. The search engine also operates to find and download the expression information for selected genes in user-selected sets of samples. In one embodiment, the DBMS 36 is an Oracle™ DBMS 36 with WebDB, which is a product produced by Oracle for implementing dynamic HTML (Hypertext Markup Language).

The storage system 40 can be any of a variety of systems that maintains information including, for example, a database server, a file storage system having large binary files, a legacy mini-computer or main-frame computer with storage. In one embodiment, the storage system 40 includes a relational database 44 in which the information is stored in a relational format. The relational database 44 includes tables of columns and rows for holding the information stored in the database 44. Each table has a primary key that is any column or set of columns storing a value or values which uniquely identify the rows in that table. The tables of the relational database 44 can also include a column or set of columns that function as a secondary key. The values of secondary key columns are used to match the primary key values of another table. The relational database 44 supports a set of operations that are performed on the relations within the database 44.

Implementation of the relational database 44 of the storage system 40 can be accomplished in various ways. For example, one embodiment of the relational database 44 is an Oracle ™ database. An example of another embodiment of the relational database 44 is a Sybase™ database.

The network 30 can be a local-area network (LAN), an Intranet, or a wide area network (WAN) such as the Internet or the World Wide Web. A user of the client 10 can be connected to the network 30 through a variety of connections including standard telephone lines, LAN or WAN links (e.g., T1, T3, 56kb, X.25), broadband connections (ISDN, Frame Relay, ATM), and wireless connections. The connections can be established using a variety of communication protocols (e.g., HTTP, TCP/IP, IPX, SPX, NetBIOS, Ethernet, RS232, and direct asynchronous connections).

During operation, the client 10 launches the browser software 12 and connects to the server 20 by specifying a resource locator corresponding to the server 20. The resource locator is specifically referred to as a Uniform Resource Locator (URL), but any type of address scheme that defines a path to a resource on the network 30 can be used to practice the principles of the invention. In response, the Web server 32 of the server 20 sends a document or Web page 38 to the client 10. In one embodiment, the Web page is written in HTML. Other document types (e.g., XML, SGML) can be used to practice the principles of the invention. An initial Web page 38 may cause the browser software 12 to prompt the user to log on by supplying a username and a password. Proper response by the client user can establish an authenticated session between the browser 12 and the server 20. Such authentication can be required before the client user is granted access to the information stored in the storage system 40.

Display of the document 38 on the screen of the client 10 produces a graphical user interface 14 that the client user can use to formulate his or her requests for access to storage system 40. The graphical user interface 14 includes one or more fields for receiving user-specified terms. To enter the terms, the user can click upon the fields with the mouse 42 and type in the terms using the keyboard. The document 38 can also include embedded hyperlinks pointing to other documents on the server 20 or on servers elsewhere on the network 30. In an alternative embodiment, the document 38 presents a line mode interface at the client 10 through which the client user submits commands, e.g., using the SQLPLUS™ tool produced by Oracle.

The Web browser 12 formats and transmits the client requests to Web server 32, which passes the request to the Web interface 34. The Web interface 34 of the server 20 converts the requests to queries in a database language (e.g., SQL). The database management system 36 of the server 20 uses the queries to access the relevant information stored in the database 44 and returns that information to the server 20 in an appropriate format. The Web server 32 then

generates a new document 38 containing the database information and transmits the new document 38 to the client 10 where the database information is displayed in the graphical user interface 14.

Fig. 2 shows an embodiment of a process for accessing information in the database 44 according to the principles of the invention. The client user uploads (step 100) raw data into the database 44. In one embodiment, the data is genomic data. Other types of data can be used to practice the principles of the invention. The raw genomic data is obtained from "chips" (or "arrays"). A chip is a solid substrate with DNA probes that are either synthesized or spotted onto the substrate surface in a grid layout. Chips may contain from a few hundred to tens of thousands of probes, each of which corresponds to a single nucleotide sequence of interest. A nucleotide sequence in turn corresponds to a genetic feature of interest, such as the coding for a specific protein. For example, a probe may refer to a mRNA strand that codes for a specific protein or amino acid sequence. Other non-mRNA probes are also placed on chips, so a nucleotide sequence may refer to a region upstream of a gene, or to a mitochondrial mRNA or other genetic material. For example, the Affymetrix GeneChip™ platform determines raw genomic data as the average difference score and present call (i.e., a measure of the presence or absence of a message) for each probe set on the array. In one embodiment, multiple measurements per spot, including the average intensity and background values for each set of probes on the array, are supported.

As used hereafter, a data set includes the genomic data that are obtained from the hybridization of one sample to a set of chips that span the genome of the organism (or some subset of the genome). A sample refers to a colony of cells grown from a particular genetic strain of organism (e.g., yeast) that has a particular genotype. Thus, the database services of the invention handle each sample independently.

Each sample is subjected to a particular treatment, which is an action taken to perturb the sample. The sample also can have a time of treatment associated with it. An experiment is a set of control and test samples and the analysis that has been applied to such samples. Often some of the hybridizations are repeated for quality control purposes. Thus, an experiment testing the effects of a single treatment may contain many samples. Other experiments study the dynamics of the treatment effects and thus entail a time course with samples corresponding to each time point measured.

In step 104, the raw genomic data is rescaled (step 104). The rescaling of the raw genomic data, described in more detail below, enables data sets for different chips to be analyzed together. The client user chooses (step 108) a reference set for the rescaled data. A reference set is a set of samples that have been normalized using the same parameters, generally with respect to one sample. Rescaled samples within the same reference set can be directly compared in an "analysis." Samples can be rescaled to multiple reference sets to enable comparisons between disparate sets of rescaled samples. This allows different rescaling decisions to be made (e.g., control-based vs. bulk-signal based, different flooring values, etc.), but still provides the option of making fast comparisons across large segments of the database 44.

For example, for control-based rescaling, foreign RNA species are added to the sample RNA in known quantities as a control for starting material. Probes are present on the chip for these foreign RNAs, so their signals can be compared from one chip to another in order to deduce proper rescaling constants. As another example, for bulk signal normalization the total signal of all probes on the chip (or some large subset of probes) are summed or averaged. This sum or average is compared between chips. This technique is only for comparisons of chips of the same type. Furthermore, if large changes in expression occur to reduce the overall signal, then this technique may be ineffective.

The Web interface 34 produces (step 112) a sample set using the rescaled samples. A Web application of the Web interface 34 performs (step 116) a user-specified analysis on the sample set. As described in more detail below, one embodiment offers two types of analysis: (1) rule based analysis, and (2) non-hierarchical clustering analysis.

Execution of the user-specified analysis produces a result (hereafter "sample result"). In one embodiment, the sample result is a list of genes (i.e., a "gene list") that are co-expressed in some way. An exemplary representation of a list of genes is:

Sample Result:
gene 1
gene 2
gene 3

In another embodiment, the sample result is a list of lists of genes (i.e., a list of gene lists). An exemplary representation of a list of lists of genes is:

Sample Result:
Gene List for Result Type 1
gene 1
gene 2
Gene List for Result Type 2
gene 3
gene 4

In still another embodiment, the sample result is a set of bit vectors. An exemplary representation of a set of bit vectors is:

Sample result:

	Result Type 1	Result Type 2	Result Type 3
gene 1	x	x	
gene 2	x	x	x
gene 3		x	
gene 4		x	x

Other embodiments of a sample result also include information that is associated with the genes in the gene list. For example, each gene can be associated with a scalar value representing a confidence metric for that gene (e.g., a scalar value of 1 means information about the gene is

present; 0 means no information about the gene is present). Accordingly, an embodiment of the sample result includes the list of genes and the scalar value associated with each gene. As another example, a sample result produced by a clustering analysis (described below) may produce a list of centroids associated with the list of genes and a graph representing a network of relationships among the genes. For this example, the sample result includes the list of centroids and the graph in addition to the list of genes. These embodiments of sample results are simply illustrative, and are not intended to limit the variety of embodiments of sample results that can be used to practice the principles of the invention.

Rule-based analyses generate results containing genes that were "up" or "down" according to certain criteria. For example, genes in a list of genes accorded an "up" result had a confidence level of present in at least one replicate of both the control and test samples, and showed a >=2 relative change in expression from control to test, with a absolute difference of at least 100 intensity points. As another example, a list of genes identified as a "down" result is similar to an up result, but the relative difference was in the downward direction (i.e. <=.5).

Other examples of sample results include "appeared" and "disappeared." Results referred to as "appeared" contain those features whose expression level was marked absent in all control samples, but present in all test samples. The expression levels of such genes are those that went from undetectable to detectable. Results referred to as "disappeared" contain those genetic features whose expression level was marked present in all control samples, but absent in all test samples.

The sample result is stored (step 120) in the database 44. The client user (or any other client user that accesses the database, whether through the same client 10 or a different client system) can associate (step 124) the stored sample result with a prior result. In one embodiment,

this association is a comparison between the stored sample result and a prior result. The comparison in one embodiment looks for genes that appear in both the stored sample result and the prior result.

The prior result can be another sample result derived from a previous analysis performed on the information in the database 44 or the prior result can be a user-created or predefined list stored in the database 44. An example of a predefined list is a MIPS-generated categorization list. MIPS stands for the Munich Information Center for Protein Sequences and is a bioinformatics group that publishes various functional categorizations of genes on the Internet. The following is an example of a small portion of the functional categorizations of yeast genes published by MIPS:

TRANSCRIPTION (751 ORFs)
- rRNA transcription (100 ORFs)
- rRNA synthesis (39 ORFs)
- rRNA processing (58 ORFs)
- other rRNA-transcription activities (3 ORFs)
- tRNA transcription (82 ORFs)
- tRNA synthesis (24 ORFs)
- tRNA processing (37 ORFs)
- tRNA modification (16 ORFs)
- other tRNA-transcription activities (4 ORFs)
- mRNA transcrition (544 ORFs)
- mRNA synthesis (410 ORFs)
- general transcription activities (64 ORFs)
- transcriptional control (326 ORFs)
- chromatin modification (32 ORFs)
- mRNA processing (splicing) (91 ORFs)
- mRNA processing (5'-, 3'-end processing, mRNA degradation) (37 ORFs)
- other mRNA-transcription activities (10 ORFs)
- RNA transport (27 ORFs)
- other transcription activities (58 ORFs)

PROTEIN SYNTHESIS (347 ORFs)
- ribosomal proteins (206 ORFs)
- translation (initiation, elongation and termination) (62 ORFs)
- translational control (30 ORFs)
- tRNA-synthetases (37 ORFs)
- other protein-synthesis activities (15 ORFs)

Each item in the MIPS list is a hyperlink to additional information regarding the functional category. For example, selecting the "other tRNA-transcription activities (4 ORFs)" hyperlink produces a Web page with the following list of genes that fall under the "other tRNA-transcription activities" category:

YOR061w	CKA2	casein kinase II alpha' chain
YOR039w	CKB2	casein kinase II beta' chain
YIL035c	CKA1	casein kinase II, catalytic alpha chain
YJL041w	NSP1	nuclear pore protein

Other examples of user-created or user-selected lists that can be stored in the database 44 are lists of chromosomes, transcription factor targets, and functional categories (e.g., metabolism genes).

Fig. 3 illustrates an embodiment of a graphical user interface 130 displayed at the client 10 upon execution of the loader 35 described in Fig. 1. The loader 35 supports file uploads from any computer system attached to the network 30 (e.g., the client 10), and provides HTTP protocol support for loading data sets from an internal web-site. Furthermore, the loader 35 allows the client user to associate the loaded data sets with information describing the experiment, such as genetic strains (in field 138), growth conditions used (in field 134), and sample treatment (in field 136). Additionally, chip lot numbers can be entered in fields 139 in order to track problems with chip and reagent quality.

In one embodiment, the loader 35 is implemented by a suite of common gateway interface (CGI) programs and modules, written in PERL, that handle the uploading of data sets to the database 44. Perl is effective for text file processing and provides a simple and well-supported database interface. It is to be understood that the loader 35 can be implemented in other ways, e.g., as an application program interface (API).

To keep data set load times to a minimum, and thus provide acceptable interactive response to the client user, the loader 35 inserts raw data row by row into an empty temporary table. The loader 35 then selects and inserts the raw data at once into a large table containing all data sets. In one embodiment, this large table contains 1.6×10^6 rows. This load optimization technique improves insert times and reduces rollback space consumption considerably. Also, the optimization technique causes insert times to be proportional to the size of the data set being inserted rather than the size of the table.

Rescaling data sets

Before data sets for different chips can be analyzed together, calibration or rescaling of the raw data in the data sets is necessary. The rescaling can be performed in a variety of ways depending on the nature of the experiment. For example, known quantities of exogenous control RNAs can be used for rescaling data values read from one chip to those read from another chip. For experiments in which the overall mRNA population is expected to remain stable, bulk signal scaling methods can also be employed. In situations where overall expression is significantly affected, for example when parts of the transcription apparatus are knocked out or inactivated because of temperature-sensitive mutations, then control-based rescaling is appropriate. Still referring to Fig. 3, the loader 35 allows the client user to choose the rescaling method (by specifying a reference set in field 135) and associated parameters when data set is loaded. The loader 35 also provides a set of default options (in field 137) that represent the typical parameters for rescaling.

To implement rescaling, a reference set is defined to include a sample used as a control for rescaling, a rescaling algorithm and any parameters that the rescaling requires, and a set of samples whose chips are rescaled to the chips from the control sample. Currently all available rescaling algorithms are stable with respect to the contents of the reference set; that is, adding

additional samples to the reference set does not affect the rescaled values of the samples already present in the reference set. Samples can be added to more than one reference set, in which case the rescaled values are stored separately for each reference set.

Not all data types are directly comparable. For example, certain measurements are only useful in a given context, while others measurements are absolute with respect to a set of experimental conditions. The rescaling of data sets occurs for just those data values that can be directly compared, as defined by the client user through the graphical user interface, but does not allow direct comparison of data values derived from different reference sets. The user assures that samples are correctly normalized (i.e., added to appropriate reference sets). If the samples are normalized correctly, then the database system constrains client users from making comparisons across reference sets, thus preventing comparisons across normalizations. Comparisons of data values derived from different reference sets occur at a higher level (e.g., during "data mining", as described below in connection with the section called "Data Mining").

Genetic namespaces

To enable querying the information in the database 44 and comparing information from different chips, probe names are "standardized" through a series of tables that map the physical probe names provided by the chip manufacturer to a unique set of genetic feature names for each organism. Accordingly, the genomic data is stored in the database 44 in two formats, the raw, unprocessed data and in a format that is optimized for analysis and querying (e.g., with genetic feature names).

This mapping of data sets into a genetic feature namespace simplifies comparisons across samples. This namespace is represented by a genetic feature table that contains one entry per genetic feature (e.g., gene, gene fragment, group of genes, or intergenic region) that is measured

by a chip probe. To map from physical chip probes to genetic features, a scheme is employed, which chooses the "best" probe on a chip for each genetic feature that is represented, based on a set of empirically chosen rules. Additionally, to make cross-technology comparisons, (e.g., from different chip manufacturers) a unique gene catalog describing every gene queried by a chip is used so that measurements of the same gene described under two different accession numbers can still be compared.

Data Retrieval

After loading and rescaling data sets, the client user can extract information from the database 44 using a retrieval tool (i.e., a Web application on the server 20) that allows the client user to select a set of genes across a set of samples, and download the resulting matrix as text or as an HTML table. The client user can load the resulting file into a spreadsheet for local (i.e., client 10) analysis.

Data Organization - Projects and Gene Categories

To organize the information stored in the database 44, the data used in analyses are divided into projects. Each project contains a sample set, which is a group of related samples derived from the same reference set. These sample sets can then be analyzed, to produce a set of results (i.e., a sample result). Each sample result can contain a list of genes or a list of gene lists and numeric values that describe that gene list, such as, for example, a centroid. Presumably the genes in a gene list are those genes that were co-expressed in an experiment. Each project is associated with an individual (e.g., a researcher). In the schema of the database 44, described below in connection with Fig. 6, each project is an entry in the PROJECTS table.

Groups of genes

Another mechanism for organizing the information in the database 44 is to place genes into user-defined categories. The categories can then be placed into groups. The MIPS functional catalogues described above is an example of this organizational mechanism. As described in more detail in the Data Mining section below, these user-defined lists of genes can be compared with lists of genes (or lists of gene lists) that are produced by user-specified analyses.

Data extraction

The manner of storage of the information in the database 44 facilitates extraction of the data sets for external analysis (i.e., local analysis) by the client user (e.g., using a spread sheet). Further, the client user can extract data sets for multiple samples across a group of features. Set operations (i.e., AND, OR, etc.) on features are also supported. For example, the set of genes up-regulated across a particular time course experiment can be combined with those genes that were down-regulated. The resulting combined set of rows can be extracted across the samples involved in the particular time course experiment or some other time course experiment for external analysis.

Data Set Analysis

To analyze the data sets stored in the database 44, the client user groups samples into sample sets. As described above, all samples in a sample set are from the same reference set, and sample sets are stored under projects for data organizational purposes. An analysis produces a comparison of the samples in the sample set to derive multiple lists of genetic features whose expression has been affected in some particular way. In a previously noted embodiment, sample sets can be analyzed using one of two tools: rule-based analysis and non-hierarchical clustering.

Rule-based analysis

Within the sample set, each sample plays a role, e.g. wild-type replica 0, time point 15' replica 1. Replicas are repeated experiments which can be used by analysis to control for experimental noise. After assigning roles to the samples, the client user chooses the rules to apply to the analysis of those samples. The client user selects the rules to apply from a set of predefined rules. The Web interface 34 then executes the selected rules in the DBMS 36 to produce a list or lists of affected genes. This sample result is then stored in the database 44, available to subsequent searches by client users.

Rule-based analysis allows the user to choose a set of predefined rules that determine which genes are co-expressed. An example of a rule is "all ORFs whose expression levels change by a factor of 2." An example of another rule is "all ORFs whose average expression levels across replicates monotonically increase over time and for which at least half of the measurements for each time point are of high confidence." Fig. 4 shows a screen shot of an exemplary graphical user interface 140 presented to the client user to perform a rule-based analysis.

In one embodiment, rule based analysis is implemented as an external module that uses R package of statistical programs, which is an implementation of the S programming language for mathematical modeling, and interacts with the database 44 through the DBMS 36. The R language is described in Ihaka & Gentleman (1996), "R: A Language for Data Analysis and Graphics", Journal of Computational and Graphical Statistics, 5, 299-314. CGI programs, written in PERL, control the R programs to provide a graphical user interface. Analyses written in R can extract a matrix of values from the database 44 corresponding to expression levels across a sample set, and determine which genetic features are co-regulated. The R programs directly load the results of the rule-based analysis in the database 44.

Cluster analysis

Cluster analyses, in general, allow the detection of patterns in gene expression without requiring previous knowledge about what those patterns should look like. After defining a sample set, the client user can export the data of the samples in the sample set and employ a variety of analysis tools to detect such patterns. An example of a type of analysis tool applies a self-organizing map algorithm to cluster genes. One such analysis tool is called GENECLUSTER, which is software produced by Whitehead Institute Center For Genome Research of Cambridge, Massachusetts. Other analysis tools can be used to analyze the sample set.

The analysis tool then uploads the output files resulting the analysis to the database 44. Such output files are then stored in the same particular format (e.g., a list of genes) as results produced by rule-based analysis. For example, the resulting cluster and associated centroids (i.e., average expression profiles) produced by an analysis tool are returned to the database 44 for further analysis as described below in the Data Mining section. A feature of the invention is that the results produced by the analysis tool are stored in the particular format to enable the comparison of results produced by different analyses irrespective of the type of analysis used. This particular format allows the addition of various programs serving as analysis tools without modifying the underlying database structure.

Data Visualization

After analyzing a sample set, client user can browse the resulting list(s) of genes associated with the analysis and their expression levels through the execution of a Java applet. The Java applet plots intensity levels or intensity fold changes using color display and produces simultaneous visualization of the expression levels of numerous genes. A fold change refers to relative change in expression of an mRNA between treated vs. untreated (or mutant vs. wild-

type) cells. It is reported as a positive number if the ratio is >=1, and as the negative reciprocal of the ratio if it is <1. Additionally, the R package of programs provides a set of plotting tools for visualizing the data. For example, some R programs plot histograms of log fold changes between chips or samples.

Data Mining

The above-described analysis and visualization tools allow client users to seek answers to questions involving a small number of samples. In accordance with the principles of the invention, the client user can also seek answers to questions that encompass different data sets or the entire database 44. As described below, the ability to compare different lists of genes provides a data mining capability.

As described above, sample results are stored in the database 44 as a set (i.e., list) of genes. Consequently, any user of a client connected to the server 20 can browse and search search through results produced by the analyses of other client users. Such searches for genes by name, strain, sample, condition, or by gene membership. For example, a client user can obtain answers to queries such as "what analyses showed a change in expression for gene X".

After the sample results are stored into the database 44, the client user can also compare those sample results with other previously stored sample results. Further, such stored sample results can be compared with other lists of genes, for example, user-defined gene lists or literature-derived classifications of genes, such as the MIPS functional catalogues. This capability enables the comparison of sample results to external information, such as knowledge extracted from scientific literature. The client user can categorize such knowledge based on whatever criteria they choose. These user-defined categorizations have a particular format adapted to facilitate comparisons with sample results stored in the database 44. The particular format follows a semi-hierarchical scheme for representing information, such as the MIPS

classifications, by function and structure. When comparing a sample result with a categorization, the sample result is considered to be a list of co-expressed genes.

In one embodiment, the results of comparing a sample result with a prior result is stored in the database 44. Because the comparison of sample results to sample results is logically equivalent to a comparison between two sets, (e.g., which members (genes) of set 1 are also members of set 2, which genes are only members of set 1), the results of the comparison can be stored in the same relational tables used to store a prior result. Thus, either bit vectors or lists of lists implemented relationally can be used.

For example, if a first sample result includes gene1 and gene2, and a second sample result includes gene1 and gene3, then a comparison of the first and second sample results produces a third result that includes gene1 (i.e., the intersection of the two sample results). This third result can then be stored as an entry in a table, just like the first and second sample results.

Fig. 5 shows an embodiment of a graphical user interface 150 presented to the client user from the server 20 for making associations between sample results and/or user-selected or user-generated gene categories. Through this interface 150, the client user perform searches across the entire database 44 for data sets that exercise particular genes or for identifying correlations between functions and expressions. The interface 150 contains two sections 152 and 154. Each section 152, 154 has a first graphical box 156, 156' in which to specify a prior result and a second graphical box 158, 158' in which to specify a sample result. The client user selects one of the two boxes 156 or 158 (and 156' or 158') in each section 152, 154, respectively. A drop down menu appears for each box 156, 156', 158, 158' presenting a menu of prior results or sample results that are available in the database 44. The client user selects the desired prior result or sample result from this menu, and the associated description of the selected prior result

or sample result appears in the respective box. Accordingly, the client user can initiate one of three types of comparisons: (1) a prior result with a prior result, (2) a prior result with a sample result, and (3) sample result with a sample result. Upon selecting the "Submit Query" button 160, a comparison is performed between the two selected results.

Examples of queries that the client user can attempt to answer through the interface 150 are "which genes that are up-regulated under condition X encode for members of the ribosomal complex?" and "which conditions show considerable overlap with enzymatic activity Y?" Such data mining queries involve set comparisons and are implemented as partially constrained Cartesian products in SQL.

Fig. 6 shows a schema 200 representing an embodiment of an organization of the database 44. The schema 200 includes tables, one or more attributes in each table, and relationships between the tables (identified by arrows between the tables). Attributes that are primary keys are underlined. The tables shown and the attributes listed under each table are not intended to be exclusive. The schema 200 can include other tables and table attributes to implement the principles of the invention.

As shown, the schema includes a SAMPLE_ON_CHIP table 202, a TSV_FILES table 204, and a TSV_RAW table 206. The SAMPLE_ON_CHIP table 202 has a Sample_ID attribute and a File_ID attribute for associating a sample of raw data with a file. The File_ID operates as a secondary key that points to the TSV_FILE table 204. The TSV_Raw table 206 stores raw data values associated with a data set. An attribute of the TSV_Raw table 606 is the File_ID, which also points to the TSV_Files table 204. The TSV_FILES table 204 includes one row corresponding to each data set loaded in the database 44 and the TSV_RAW table 206 contains one row for each probe present in the data file.

The schema also includes a SAMPLES table 208, a GROW_CONDITION table 210, and a STRAIN table 212. The SAMPLES table 208 includes a CONDITION_ID attribute and a STRAIN_ID attribute that associate each sample in the table 208 with a growth condition and a strain, respectively. The CONDITION_ID attribute operates as a secondary key for searching the GROW_CONDITION table 210, and the STRAIN_ID attribute operates as a secondary key for searching the STRAIN table 212. Each entry in the STRAIN table 212 provides a description of the particular strain of organism and each entry in the GROW_CONDITION table 210 provides a description under which a strain is grown. The SAMPLES table 208 also includes a SAMPLE_ID attribute that corresponds to the SAMPLE_ID attribute of the SAMPLE_ON_CHIP table 202.

Other tables in the schema 200 include a REFERENCE_SET table 214, a SAMPLE_IN_REFERENCE_SET table 216, a ABS_EXPRESSION table 218, and a ABS_DATA_TAB table 220. The REFERENCE_SET table 214 groups samples that have been rescaled together using the same set of parameters and a single control sample. Each sample other than the control sample is rescaled using parameters and the values associated with the control sample. The SAMPLE_IN_REFERENCE_SET table 216 maintains the relationships between samples and reference sets. The SAMPLE_IN_REFERENCE_SET table 216 includes a Reference_set_ID attribute that is a secondary key for searching the REFERENCE_SET table 214 and a Sample_ID attribute that points to the SAMPLES table 208.

The ABS_EXPRESSION table 218 stores an entry for every chip that is inserted into a reference set. Attributes of the ABS_EXPRESSION table 218 store information describing the rescaling, such as scale factor and reference chip. The ABS_DATA_TAB table 220 stores rescaled data values and points to the SAMPLE_IN_REFERENCE_SET table 216.

Still other tables in the schema 200 include a SAMPLE_SET table 222, an ANALYSIS_RESULTS table 224, a GENE_IN_LIST table 226, a PROJECTS table 228, a SAMPLE_IN_PROJECTS table 230, a SAMPLE_IN_SSET table 232, and an ANALYSIS_PARAMETERS table 234.

The SAMPLE_SET table 222 groups samples that are analyzed together. In one embodiment, all samples in a sample set come from the same reference set. The ANALYSIS_RESULTS table 224 holds the sample result sets generated by an analysis. There is one entry in the ANALYSIS_RESULTS table 224 for each sample result produced by an analysis. Note that one analysis may produce multiple gene lists (thus, the sample result is a list of gene lists). The ANALYSIS_PARAMETERS table 234 identifies the parameters used to

perform a given analysis. There is one entry in the ANALYSIS_PARAMETERS table 234 for each analysis performed. The GENE_IN_LIST table 226 joins the sample results with the genetic features such results contain. There is one entry in the GENE_IN_LIST table 226 for each gene identified in a sample result.

As described above in the Data Organization section, the PROJECTS table 228 holds projects which is an organizational construct that includes an arbitrary group of samples and the sample sets derived from such samples. Indirectly, the PROJECTS table 228 groups analyses. The SAMPLE_IN_PROJECTS table 230 includes one entry per sample in a project. The SAMPLE_IN_SSET table 232, which includes one entry per sample in a project in sample set, associates samples in projects with sample sets.

Example of Operation

The overall operation of the invention is illustrated by the following example. In particular, this example demonstrates how a client user loads raw genomic data into the database 44, generates sample results from genomic data in the database 44, and performs data mining by associating the stored sample results with other previously stored sample results and a user-selected or user-generated list of genes.

Consider the following experiment conducted on two genes, YOR095C ("RKI1") and YFL014W ("HSP12") across four samples: two control samples and two deletions of "cse2/med9." Assume that RKI1's expression drops by more than two-fold in this experiment, and HSP12 increases by at least two-fold.

Data is loaded from one data file per array, produced by scanning software. Referring back to Fig. 3, the graphical user interface 130 presented to the client user includes fields 132 for identifying the data files from which to load the data and for associating sample information with

that data. In this example, there are four data files, one for each of the four chips associated with one sample. Each data file contains one or more measurements of interest per probe located on array. The loader 35 uploads each data file into multiple tables, including the TSV_RAW 206 and TSV_FILES 204 tables. The TSV_FILES table 204 then contains one row for each data set loaded. The TSV_RAW table 206 contains one row for each probe present in the data file, as shown for example in TABLE 1 below:

TABLE 1

TABLE NAME TSV_RAW			
FILE_ID	PROBE_ID	AVGDIFF	CONF
100	200	374	P
101	201	258	P

Using the SAMPLE_ON_CHIP table 202, the data set is associated with sample information describing the sample and the chip (array) on which the sample was hybridized, as shown in TABLE 2 below:

TABLE 2

TABLE NAME SAMPLE_ON_CHIP		
SAMPLE_ID	FILE_ID	CHIP_ID
300	100	400
300	101	401

Then the loaded data is rescaled with respect to a pre-defined set of rescaling parameters (reference set). The rescaling constants for each data file are stored in the ABS_EXPRESSION table 218, as shown in TABLE 3 below:

TABLE 3

TABLE NAME ABS_EXPRESSION		
RESCALED_SAMPLE_ID	CHIP_ID	FACTOR
500	400	0.60953
500	401	0.78251

The raw data is divided by the appropriate rescaling factor and stored in the ABS_DATA_TAB table 220. Using the ABS_DATA_TAB table 220, each data point is associated with the gene that the probe queries, as shown in TABLE 4 below:

TABLE 4

TABLE NAME ABS_DATA_TAB			
RESCALED_SAMPLE_ID	GENE	AVGDIFF	CONFIDENCE (present (P))
500	YOR095C	613.6	1
500	YFL014W	329.7	1

The above-described rescaling process is repeated for all samples to be compared in an analysis. Referring now to Fig. 4, the type 141 of analysis is chosen, here static analysis ("SA"), and the rescaled samples 142 to be analyzed together are selected to define a sample set. Static analysis is an appropriate analysis for systems in equilibria (e.g., knockouts, deletions, mutations). The rescaled samples 142 are collected into SAMPLE_IN_SAMPLE_SET 232 as shown in TABLE 5 below. All samples in the sample set are derived from the same reference set.

TABLE 5

TABLE NAME SAMPLE_IN_SAMPLE_SET		
ID	SAMPLE_SET_ID	RESCALED_SAMPLE_ID
600	700	500
601	700	501
602	700	502
603	700	503

The type 144 and replicate 146 fields are used to structure the comparison between the samples in the sample set. For the static analysis of the present example, samples of sample type "WT" (i.e., wild type) are compared against samples of sample type "MT" (i.e., mutant type). So, replicate samples are compared against samples of the same replicate, e.g., replicate 1 ples are compared against replicate 1 samples, and replicate 2 samples are compared against

replicate 2 samples. Various other types of comparisons are possible, For example, another method for comparing the samples in the sample set is to average the mutant replica values and to divide that average by the average of the wildtype values.

The selected analysis is performed and the sample results are stored. In this example, the analysis performed compares the average expression level of the control samples to that of the test samples for each gene, determining if the genes differ by more than a factor of 2 either up or down. If the test samples are at least 2 times (2X) the control samples, the gene is assigned to the "up" result. If test samples are at least 2X lower, then the gene is assigned to the "down" result. Referring to TABLE 6 below, the selected analysis (here, ANALYSIS ID 900) illustrates an example of an analysis that can produce multiple lists of genes (i.e., a list of lists): one list for "up" genes, and another list for "down" genes.

TABLE 6

TABLE NAME ANALYSIS RESULTS		
ID	ANALYSIS_ID	NAME
800	900	up
801	900	down

As shown in TABLE 7, the GENE_IN_LIST table 226 associates each gene with the appropriate result(s) for that gene:

TABLE 7

TABLE NAME GENE_IN_LIST	
GENE	RESULT_ID
YOR095C	801
YFL014W	800

Now answers to questions such as "which genes were in result "up" in analysis x and in analysis y" can be provided by the database 44. In the present example, the gene YFLO14W is a gene with an "up" result.

Also, exhaustive searches such as "find two sets, X and Y, such that set X <> set Y and {X} = {Y}" (i.e., find any two overlapping sets) can be performed. For example, searching through a filter set of user-defined sets (e.g., the MIPS categories) for the gene YFL014W, one may find the gene YFL014W in the groups shown in TABLE 8. Table 8 is a subset of rows in a relational table containing categories that include the gene YFL014W.

TABLE 8

ID	GROUP_ID	U_DESC	EXT_ID	ID_1	UCAT_ID
551	286	Signal transduction	10000000	22648	551
557	286	Osmosensing	10030000	22784	557
561	286	Other osmosensing activities	10039900	22805	561
575	286	Cell rescue, defense, cell death and aging	11000000	23221	575
576	286	Stress response	11010000	23333	576
577	286	DNA repair (direct repair, base excision repair and nucleotide excision repair)	11040000	23499	577

If the selected analysis (ANALYSIS_ID = 900) determined that multiple genes were in the "up" result, then the statistical significance of an overlap of the genes in the "up" result with the genes in any of these categories could be assessed.

Fig. 7 shows an example of a display 240 at the client 10 that is produced when searching for the gene YFL014C in sample results previously stored in the database 44. The gene view shows HSP12 (YFL014C) in the result "up" for the "cse2/med9," "sin4," and "srb10" experiments 242. Additional information stored for each sample result is also shown, namely a value 243 and a graphical representation 244 of the fold change for each experiment (here, 9.35 for the cse2/med9 experiment, 11.51 for the sin4 experiment, and 32.75 for the srb10 experiment).

While the invention has been shown and described with reference to specific preferred embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the following claims.

Claims

1. A method for analyzing data over a network, comprising the steps of:

receiving data;

rescaling the data to produce rescaled data;

associating the rescaled data with a pre-selected set of parameters;

generating a sample set from the associated rescaled data;

performing analysis on the sample set to produce a sample result;

storing the sample result in a database; and

associating the stored sample result with a prior result.

2. The method of claim 1 wherein the prior result is a sample result previously stored in the database.

3. The method claim 1 wherein the prior result is a user-generated result.

4. The method claim 1 wherein the prior result is a user-selected result.

5. The method of claim 1 storing the rescaled data in the same database as the sample result.

6. The method of claim 1 wherein the stored sample result is a list of lists.

7. The method of claim 6 wherein each list in the list of lists is a list of genes.

8. The method of claim 1 wherein the stored sample result is a set of bit vectors.

9. The method of claim 1 where the associating comprises comparing the sample result with the prior result.

10. The method of claim 9 further comprising storing the results of associating the stored sample result with the prior result in the database.

11. The method of claim 1 further comprising defining a reference set to be used as a control for rescaling the provided data.

12. A system for analyzing data, comprising:

a calibrator rescaling the data;

a pre-selected set of parameters associated with the rescaled data;

a sample set generated from the associated rescaled data;

an analyzer performing analysis on the sample set to produce a sample result;

a database storing the sample result; and

an associator associating the stored sample result with a prior result.

13. The system of claim 12 wherein the prior result is a sample result previously stored in the database.

14. The method claim 12 wherein the prior result is a user-generated result.

15. The method claim 12 wherein the prior result is a user-selected result.

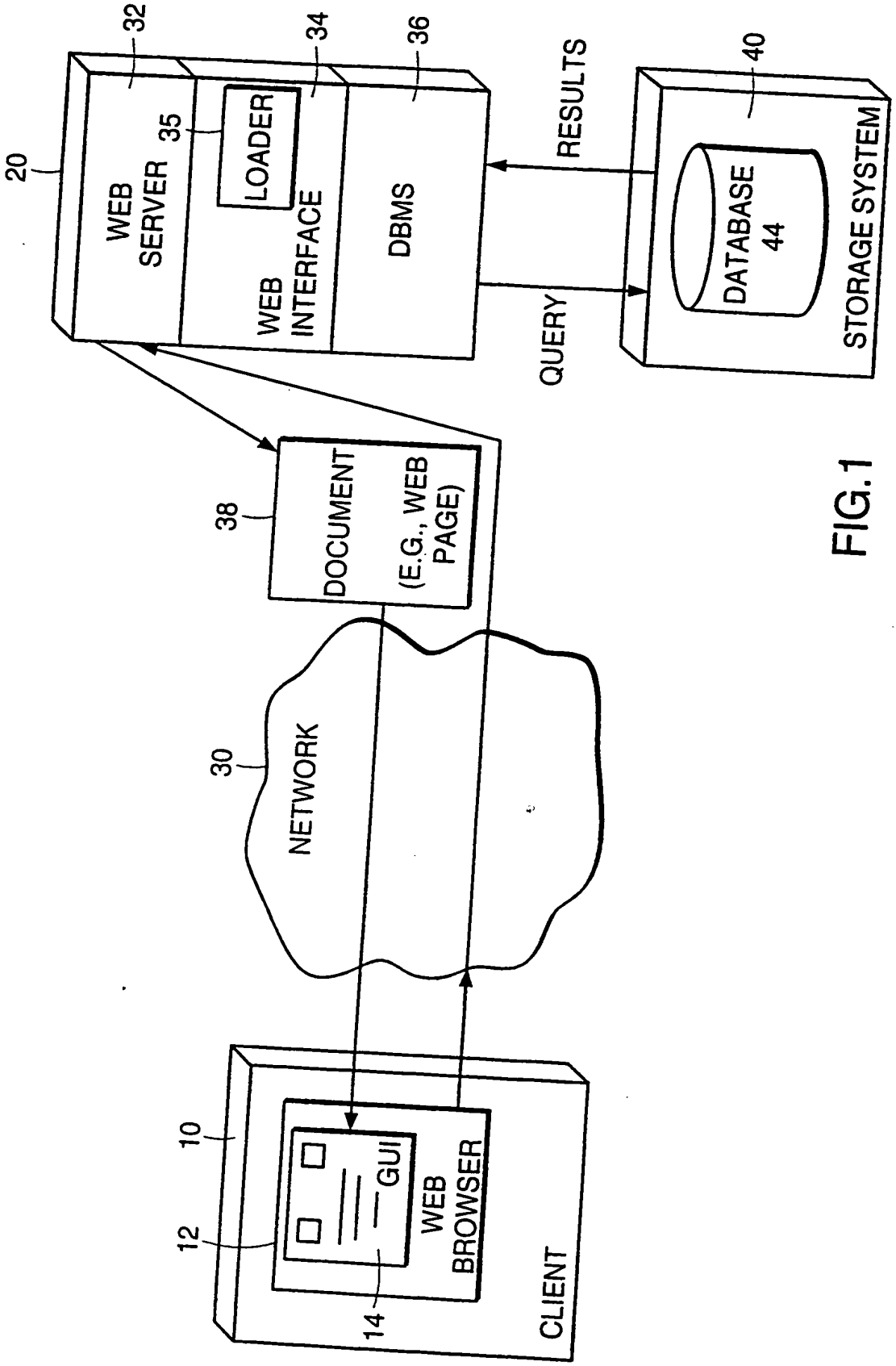

20
32
34
36
35
WEB SERVER
LOADER
WEB INTERFACE
DBMS
RESULTS
QUERY
40
DATABASE 44
STORAGE SYSTEM
38
DOCUMENT (E.G., WEB PAGE)
30
NETWORK
10
12
GUI
WEB BROWSER
14
CLIENT

FIG. 1

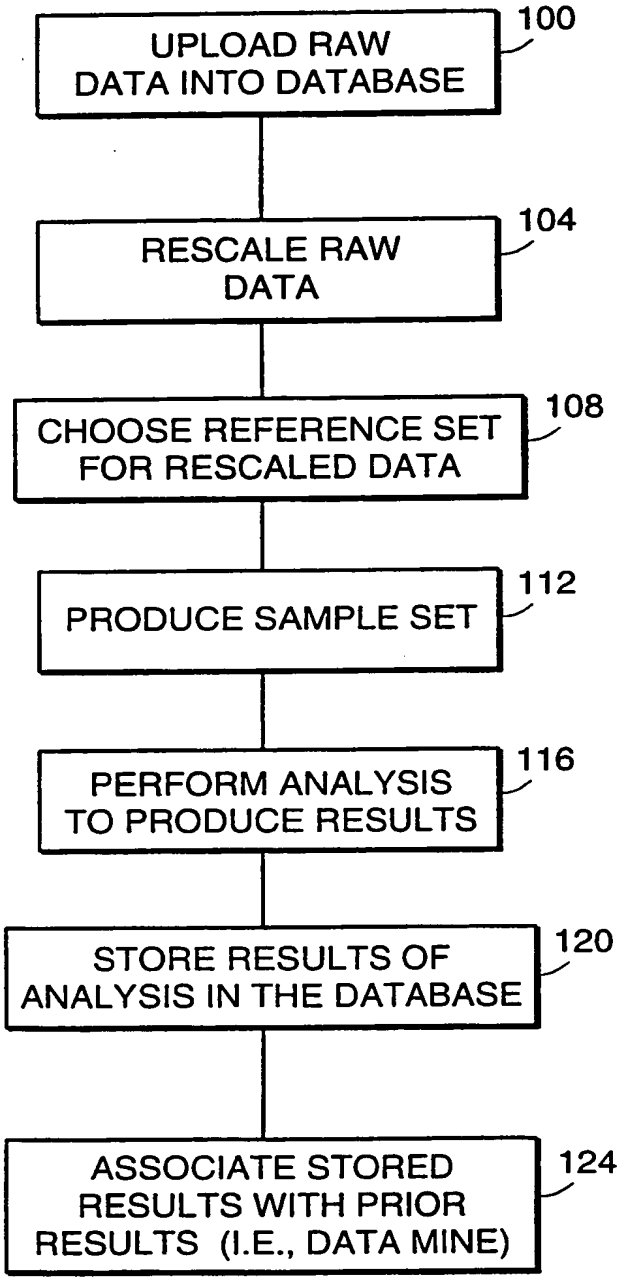


FIG. 2

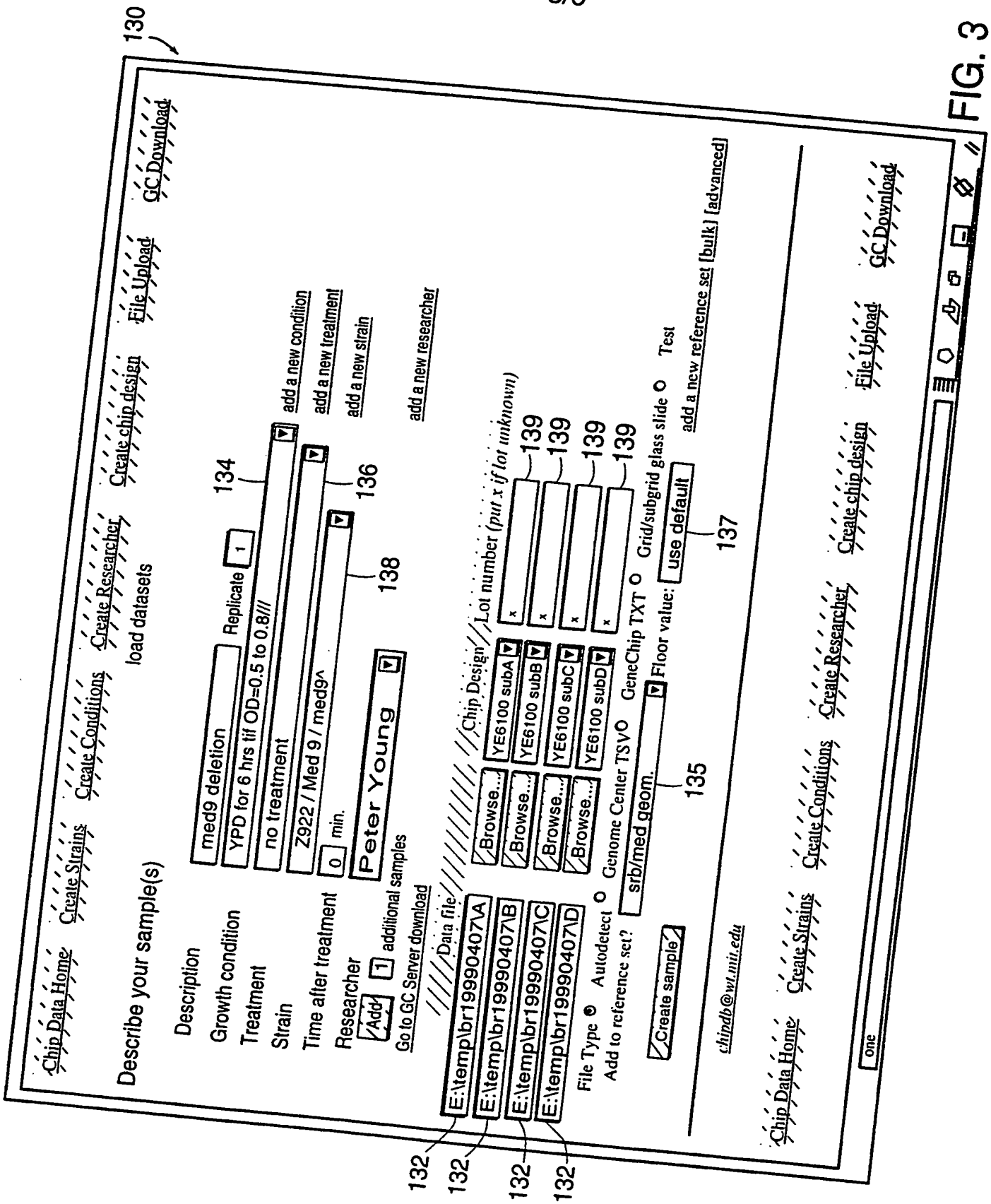

130
Chip Data Home
Create Strains
Create Conditions
Create Researcher
Create chip design
File Upload
GC Download
load datasets
Describe your sample(s)
Description
Growth condition
Treatment
Strain
Time after treatment
Researcher
Add
1
additional samples
Go to GC Server download
med9 deletion
Replicate
1
134
YPD for 6 hrs tif OD=0.5 to 0.8///
no treatment
Z922 / Med 9 / med9^
0
min.
Peter Young
138
136
add a new condition
add a new treatment
add a new strain
add a new researcher
Data file
Chip Design
Lot number (put x if lot unknown)
E:\temp\br19990407\A
E:\temp\br19990407\B
E:\temp\br19990407\C
E:\temp\br19990407\D
Browse...
YE6100 subA
YE6100 subB
YE6100 subC
YE6100 subD
x
139
File Type
Autodetect
Genome Center TSV
GeneChip TXT
Grid/subgrid glass slide
Test
Add to reference set?
srb/med geom.
135
Floor value:
use default
137
add a new reference set [bulk] [advanced]
Create sample
chipdb@wi.mit.edu
132
one

FIG. 3

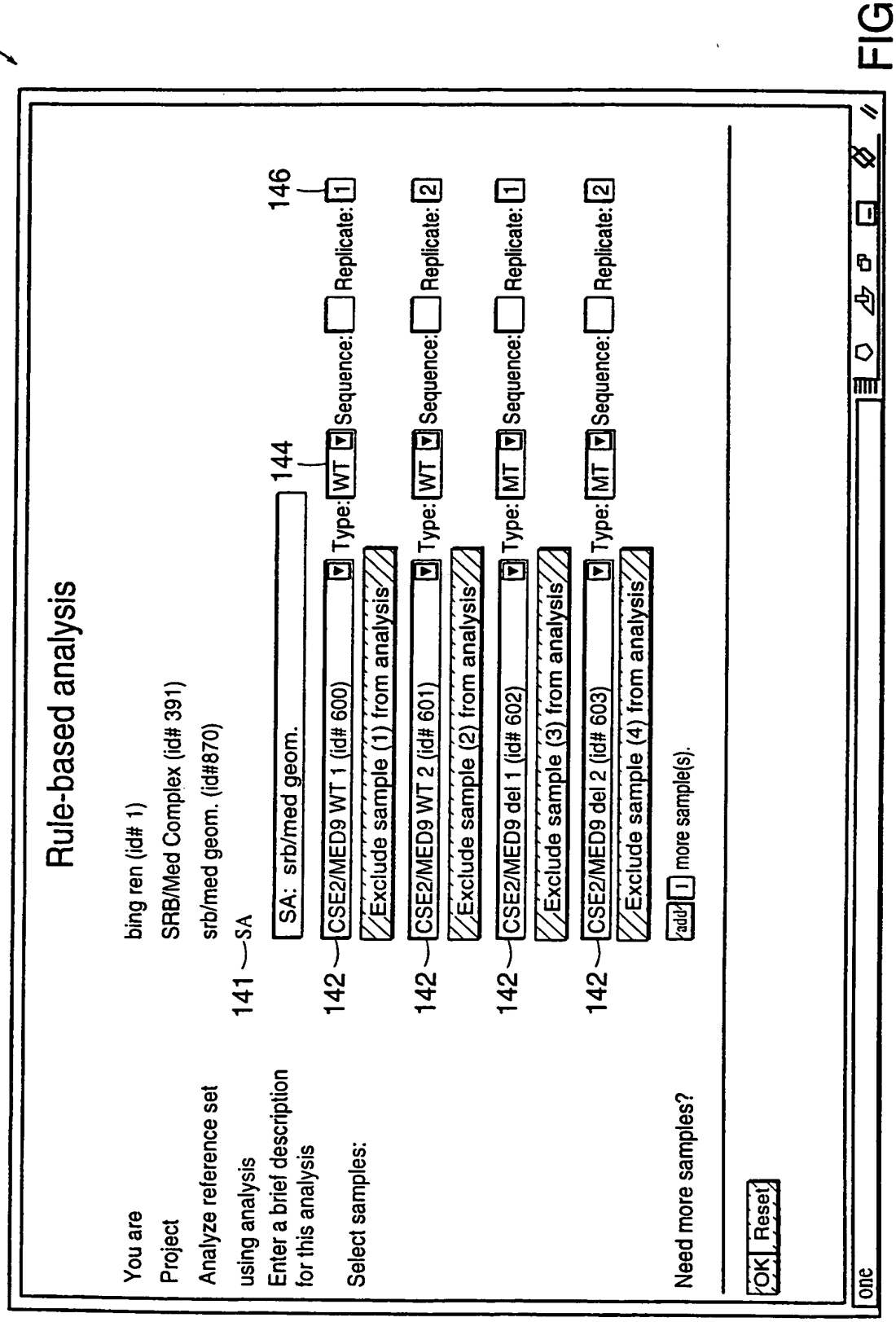


FIG. 4

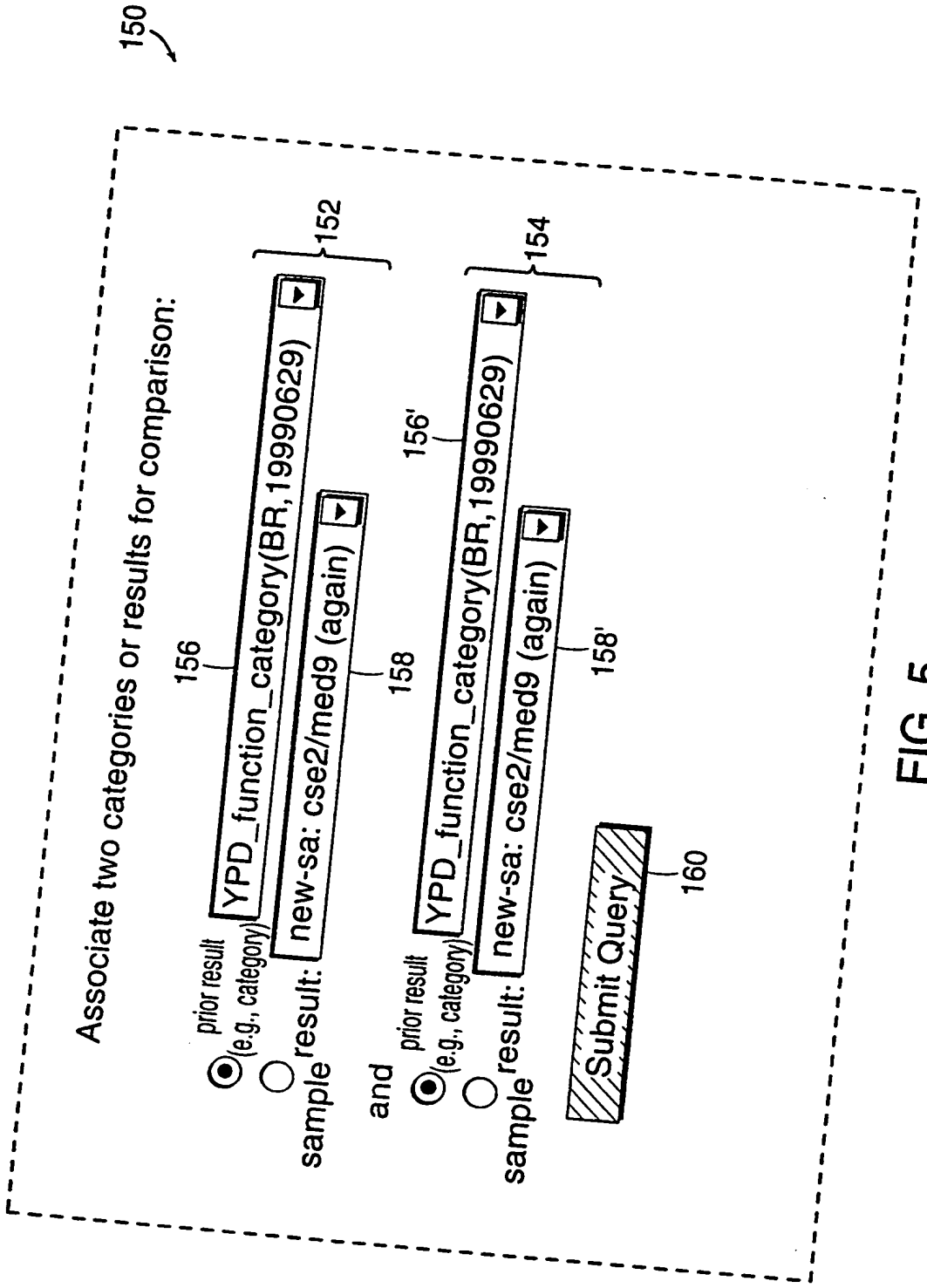

150
Associate two categories or results for comparison:
156
YPD_function_category(BR,19990629)
152
prior result (e.g., category)
sample result:
new-sa: cse2/med9 (again)
158
and
156'
YPD_function_category(BR,19990629)
154
prior result (e.g., category)
sample result:
new-sa: cse2/med9 (again)
158'
Submit Query
160

FIG. 5

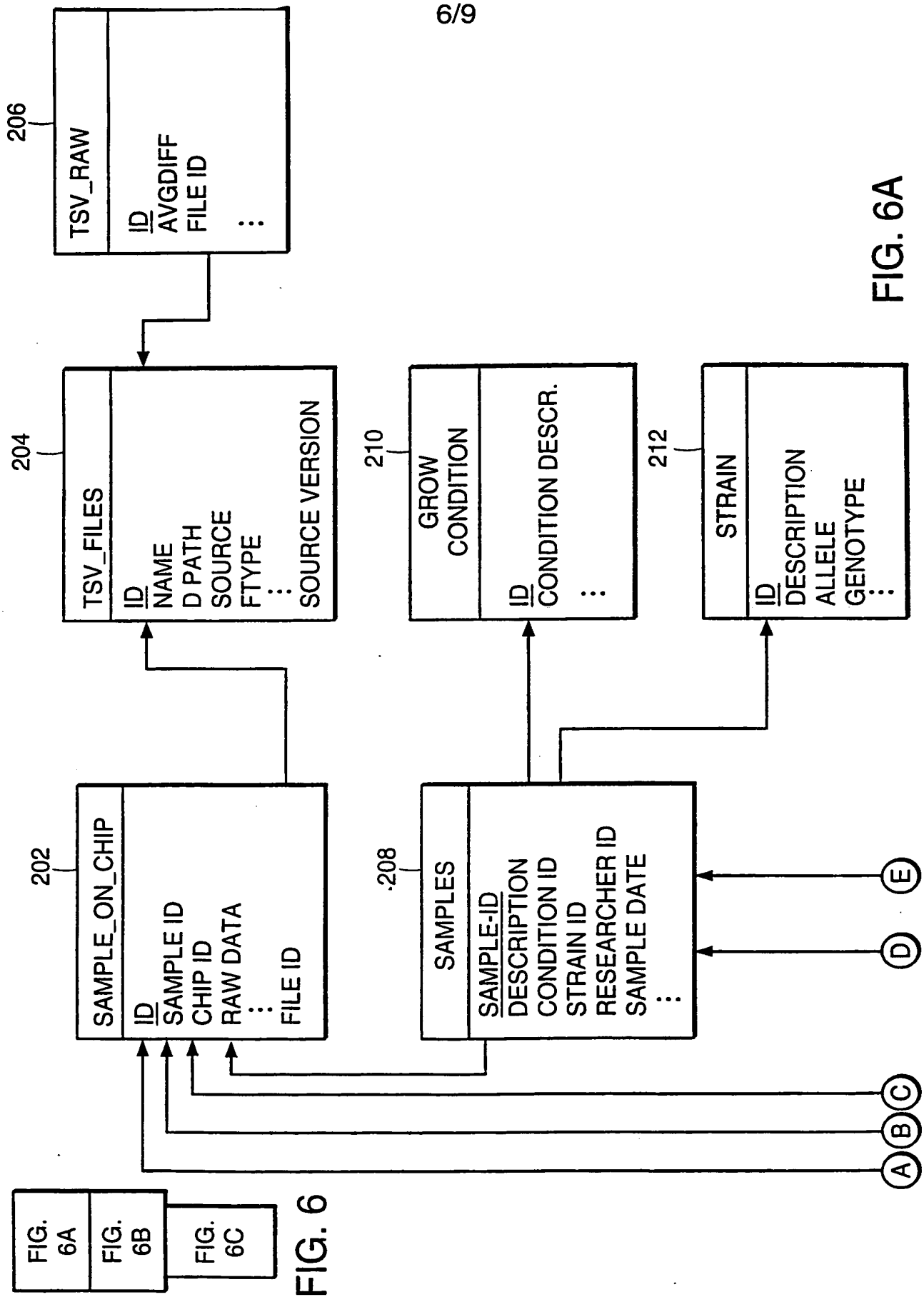


FIG. 6A

FIG. 6

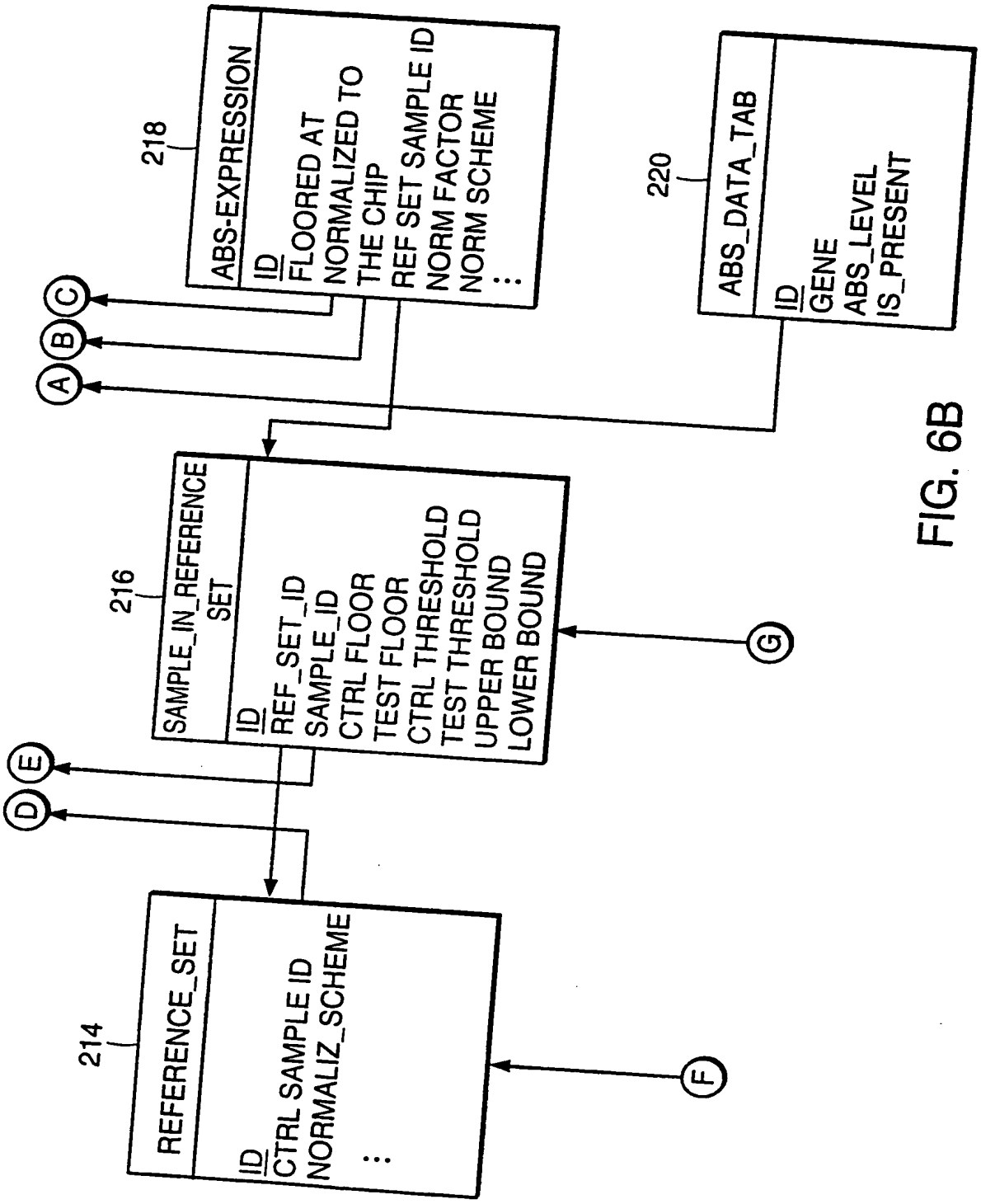

218
ABS-EXPRESSION
ID
FLOORED AT
NORMALIZED TO
THE CHIP
REF SET SAMPLE ID
NORM FACTOR
NORM SCHEME
...
220
ABS_DATA_TAB
ID
GENE
ABS_LEVEL
IS_PRESENT
A
B
C
216
SAMPLE_IN_REFERENCE SET
ID
REF_SET_ID
SAMPLE_ID
CTRL FLOOR
TEST FLOOR
CTRL THRESHOLD
TEST THRESHOLD
UPPER BOUND
LOWER BOUND
G
D
E
214
REFERENCE_SET
ID
CTRL SAMPLE ID
NORMALIZ_SCHEME
...
F

FIG. 6B

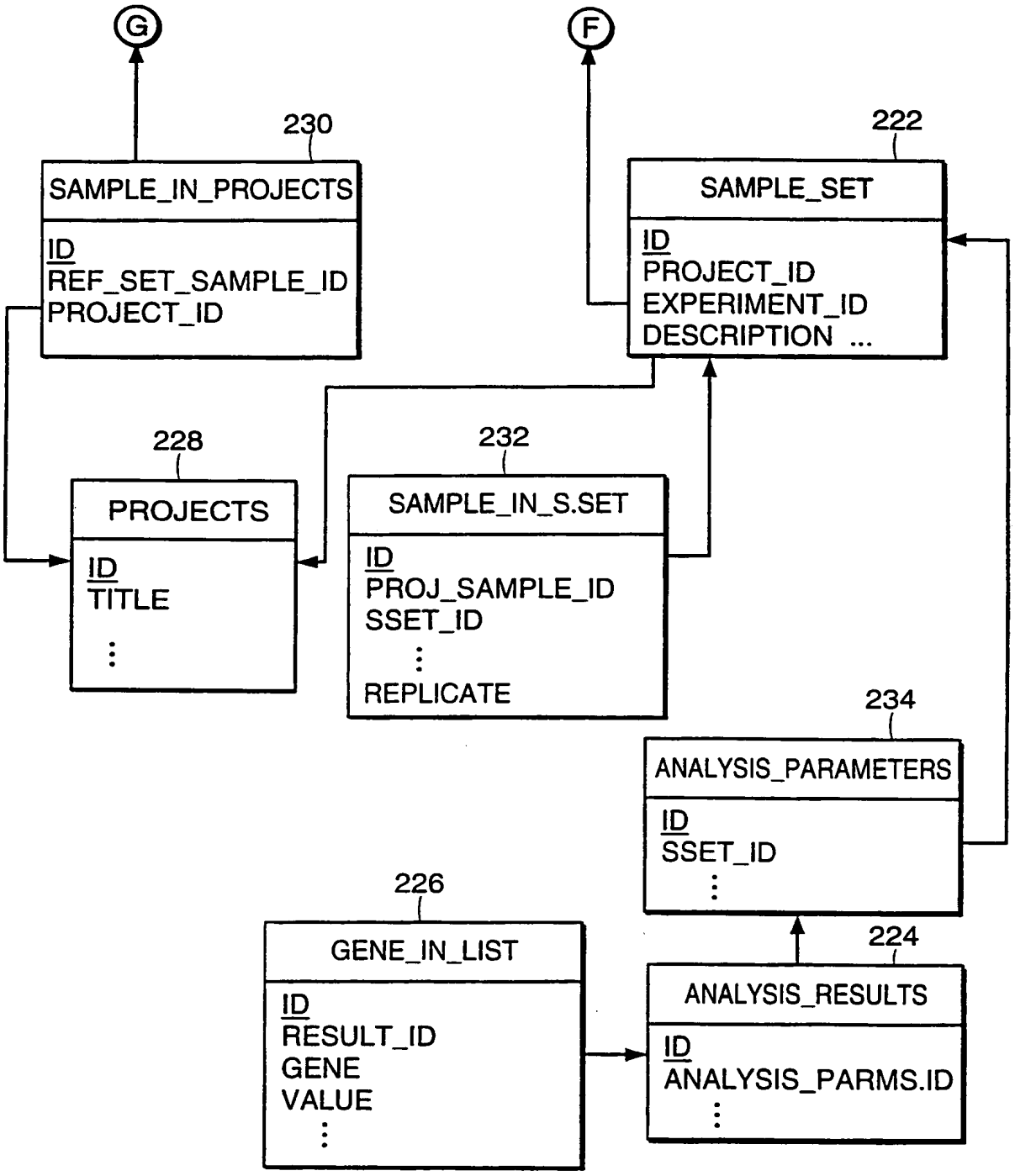


FIG. 6C

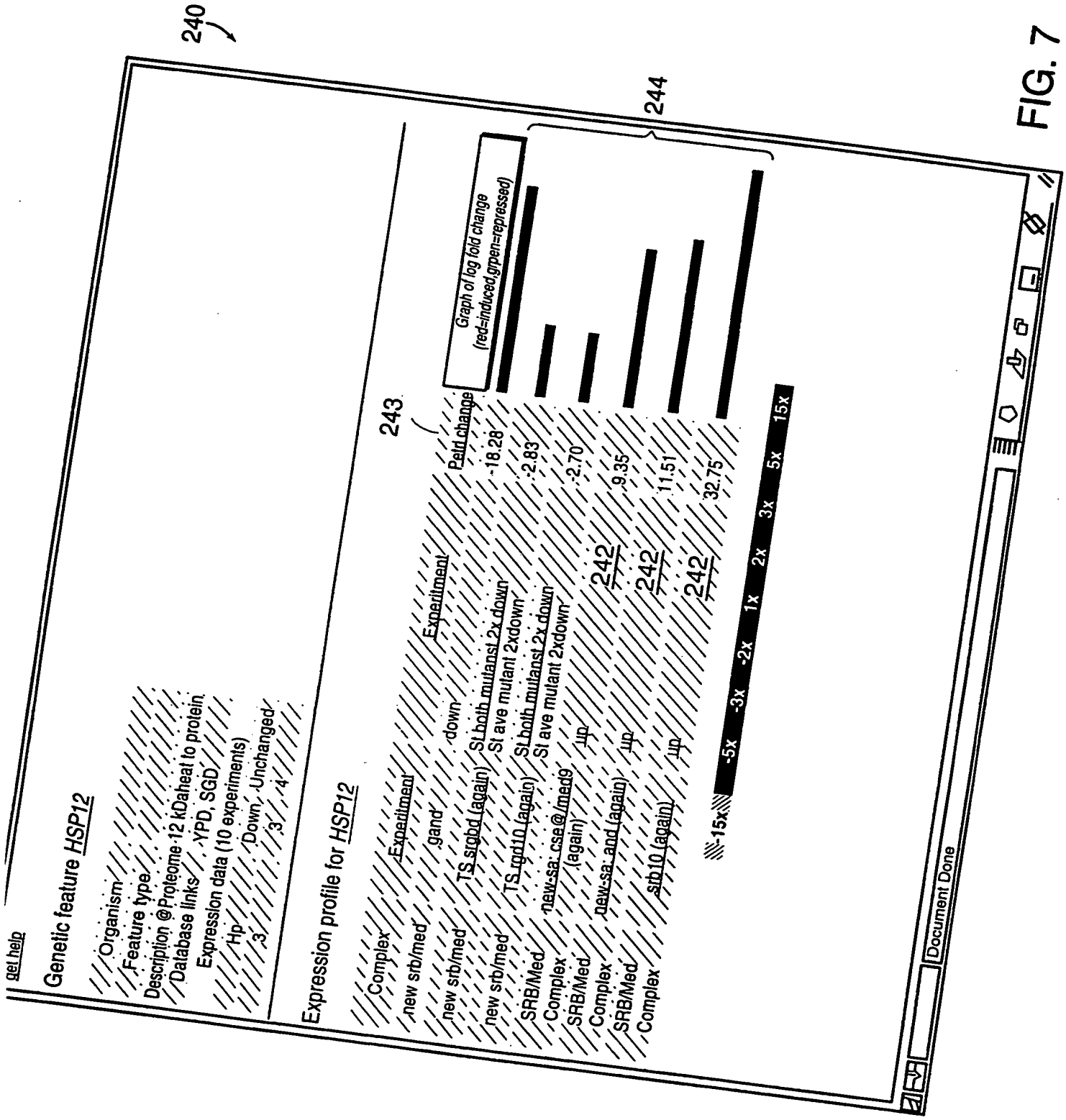

240
244
243
242
242
242
get help
Genetic feature HSP12
Organism
Feature type
Description @Proteome 12 kDaheat to protein
Database links YPD, SGD
Expression data (10 experiments)
Hp
3
Down Unchanged
3 4
Expression profile for HSP12
Complex
new srb/med
new srb/med
new srb/med
SRB/Med
Complex
SRB/Med
Complex
SRB/Med
Complex
Experiment
gand
down
TS srgbd (again)
TS rgd10 (again)
new-sa: csa@/med9
(again)
new-sa: and (again)
srb10 (again)
Graph of log fold change
(red=induced,green=repressed)
Fold change
-18.28
-2.83
-2.70
9.35
11.51
32.75
St both mutanst 2x down
St ave mutant 2xdown
St both mutanst 2x down
St ave mutant 2xdown
up
up
up
-15x
-5x -3x -2x 1x 2x 3x 5x 15x
Document Done

FIG. 7

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 November 2000 (23.11.2000)

PCT

(10) International Publication Number
WO 00/70556 A3

(51) International Patent Classification[7]: G06F 19/00

(21) International Application Number: PCT/US00/13823

(22) International Filing Date: 19 May 2000 (19.05.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/134,793 19 May 1999 (19.05.1999) US

(71) Applicant: WHITEHEAD INSTITUTE FOR BIOMEDICAL RESEARCH [US/US]; Nine Cambridge Center, Cambridge, MA 02142 (US).

(72) Inventors: REN, Bing; 39 Springfield Street, Somerville, MA 02143 (US). YOUNG, Richard; 216 Highland Street, Weston, MA 02493 (US). YOUNG, Peter; 48 Lowell Street, Somerville, MA 02143 (US).

(74) Agent: RODRIGUEZ, Michael, A.; Testa, Hurwitz & Thibeault, LLP, High Street Tower, 125 High Street, Boston, MA 02110 (US).

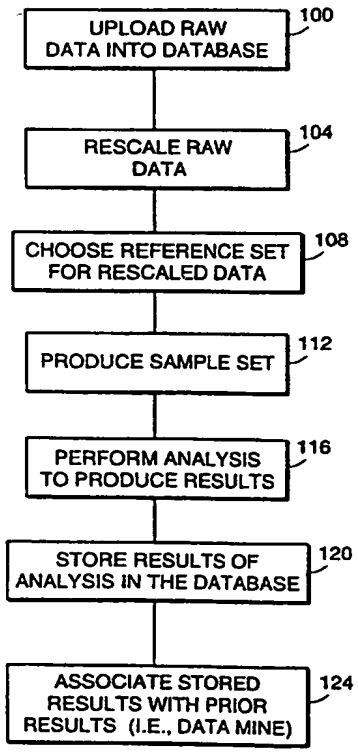
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: A METHOD AND RELATIONAL DATABASE MANAGEMENT SYSTEM FOR STORING, COMPARING, AND DISPLAYING RESULTS PRODUCED BY ANALYSES OF GENE ARRAY DATA



(57) Abstract: A method and system for analyzing data over a network are described. A Web server communicates with a storage system that stores genomic information in a database. Client systems connect to the Web server over a network, such as the Internet, using standard Web protocols (e.g., HTTP). The Web server sends Web pages to the client through which pages the user of the client can load genomic information into the database. The client user obtains the genomic information for uploading from genomic samples of organisms hybridized to chips or arrays. With the database populated with genomic information, the client user interactively selects and performs an analysis on selected samples over the network. The result produced by the analysis is a list of genes or a list of gene lists that becomes part of the database. These gene lists or lists of gene lists can then be compared with other previously stored lists or with user-generated and/or user-selected gene lists. Accordingly, subsequent users of the database can review the research performed by others, and incorporate that research into their own research.

WO 00/70556 A3

(88) Date of publication of the international search report:
16 August 2001

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

XOCID: <WO___0070556A3_I_>

INTERNATIONAL SEARCH REPORT

Inte. onal Application No
PCT/US 00/13823

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F19/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ECKMAN B A ET AL: "The Merck Gene Index browser: an extensible data integration system for gene finding, gene characterization and EST data mining" BIOINFORMATICS,GB,OXFORD UNIVERSITY PRESS, SURREY, vol. 14, no. 1, February 1998 (1998-02), pages 2-13, XP002132418 ISSN: 1367-4803 the whole document ---	1-15
X	WO 96 23078 A (INCYTE PHARMA INC ;SEILHAMER JEFFREY J (US); DELEGEANE ANGELO (US)) 1 August 1996 (1996-08-01) abstract; claims 1-20 ---	1-15
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 February 2001	02/03/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Filloy García, E

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/US 00/13823

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 692 107 A (KERBER RANDY G ET AL) 25 November 1997 (1997-11-25) abstract; figures 1-3 column 2, paragraph 4 - paragraph 5	1-6,8-15
P,X	WO 00 15847 A (STEWARD KEITH LEROY ;SHI QIN (US); GENE LOGIC INC (US); CARIASO MI) 23 March 2000 (2000-03-23) abstract; figure 1	1-15
A	WO 97 47763 A (CURAGEN CORP) 18 December 1997 (1997-12-18) page 72, paragraph 3 section 5.2.9.1.	1-15
A	SATOU K ET AL: "FINDING ASSOCIATION RULES ON HETEROGENEOUS GENOME DATA" PROCEEDINGS OF THE PACIFIC SYMPOSIUM ON BIOCOMPUTING, 6 January 1997 (1997-01-06), XP000889673	

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte onal Application No
PCT/US 00/13823

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9623078 A	01-08-1996	AU 688465 B	12-03-1998
		AU 1694695 A	15-08-1995
		AU 692626 B	11-06-1998
		AU 3759095 A	14-08-1996
		BG 100751 A	31-07-1997
		BR 9506657 A	16-09-1997
		CA 2210731 A	01-08-1996
		EP 0748390 A	18-12-1996
		EP 0805874 A	12-11-1997
		FI 962987 A	26-09-1996
		JP 9503921 T	22-04-1997
		NO 963151 A	27-09-1996
		NZ 294720 A	26-06-1998
US 5692107 A	25-11-1997	NONE	
WO 0015847 A	23-03-2000	AU 6244099 A	03-04-2000
WO 9747763 A	18-12-1997	US 6083693 A	04-07-2000
		AU 3395597 A	07-01-1998
		CA 2257958 A	18-12-1997
		EP 0912753 A	06-05-1999
		US 6057101 A	02-05-2000